

Dynamic Channel List

Overview

The goal of this assignment is to implement the list view of channels that interactively moves with animation. You may build the project using React and other third-party libraries. For the browser compatibility, the app should run correctly in the latest version of Chrome browser.

Note: You are welcome to utilize AI tools, such as ChatGPT and Claude, to enhance your code. However, if you use any AI tools, please include specific details about how you used them (e.g., the type of tool and the prompt you provided) when submitting your assignment.

Preparation

- Sendbird account is required. You can go to the [Sendbird Dashboard](#) and create an application.
- Once you setup the Sendbird app in the Dashboard, install Sendbird Chat SDK for JavaScript.

Limitations

- Use the recent version of React framework.
- Do not use **GroupChannelCollection** to manage the channel list.
- Only use the following Sendbird Chat SDK functions to achieve the goal.
 - **SendbirdChat.init**(options) with the following options.
 - **appId**: Your app ID
 - **localCacheEnabled**: false
 - **modules**: [new **GroupChannelModule**()]
 - **sendbirdChat.connect**(userId)
 - **sendbirdChat.groupChannel.createMyGroupChannelListQuery**() with the following params.
 - **includeEmpty**: true
 - **limit**: 10
 - **order**: **GroupChannelListOrder.CHANNEL_NAME_ALPHABETICAL**
 - **sendbirdChat.groupChannel.createChannel**(params) with the following params.
 - **name**: a randomly generated 8-English-letter string
 - **channel.updateChannel**(params) with the following params.
 - **name**: a randomly generated 8-English-letter string

Step 1

Create a list view of names sorted in alphabetical order. You can make a dummy 8-English-letter string array to achieve this. When the mouse hovers an item, the item should move **40px right** and the adjacent items should move **20px right**. Then they should move back when the mouse cursor is out. See Figure 1.

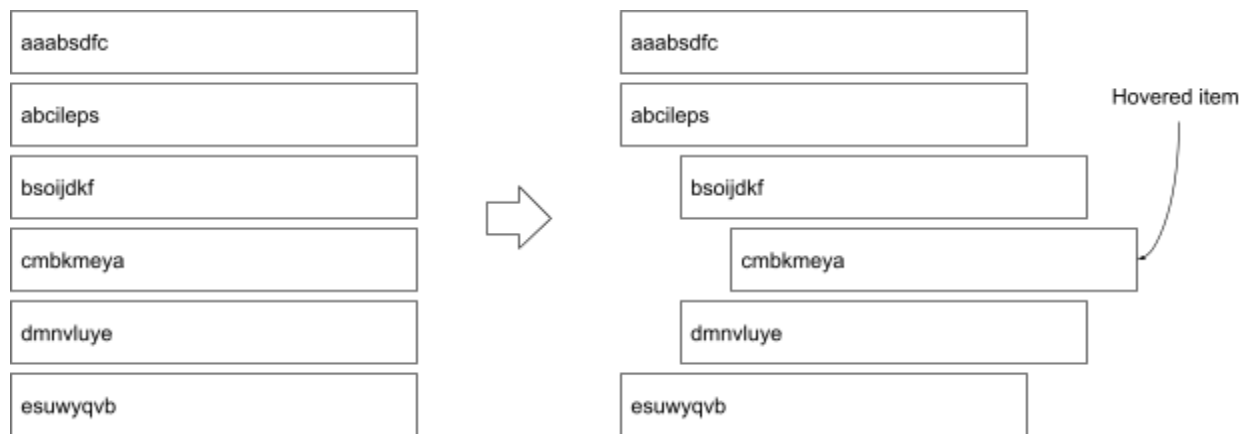


Figure 1. Animated sorted list view

Step 2

Add "Create new channel" button on the top. It should create a new channel with the name of randomly generated string with 8-English-letters. When a channel is created, locate the channel to the right place in the list view. See Figure 2.

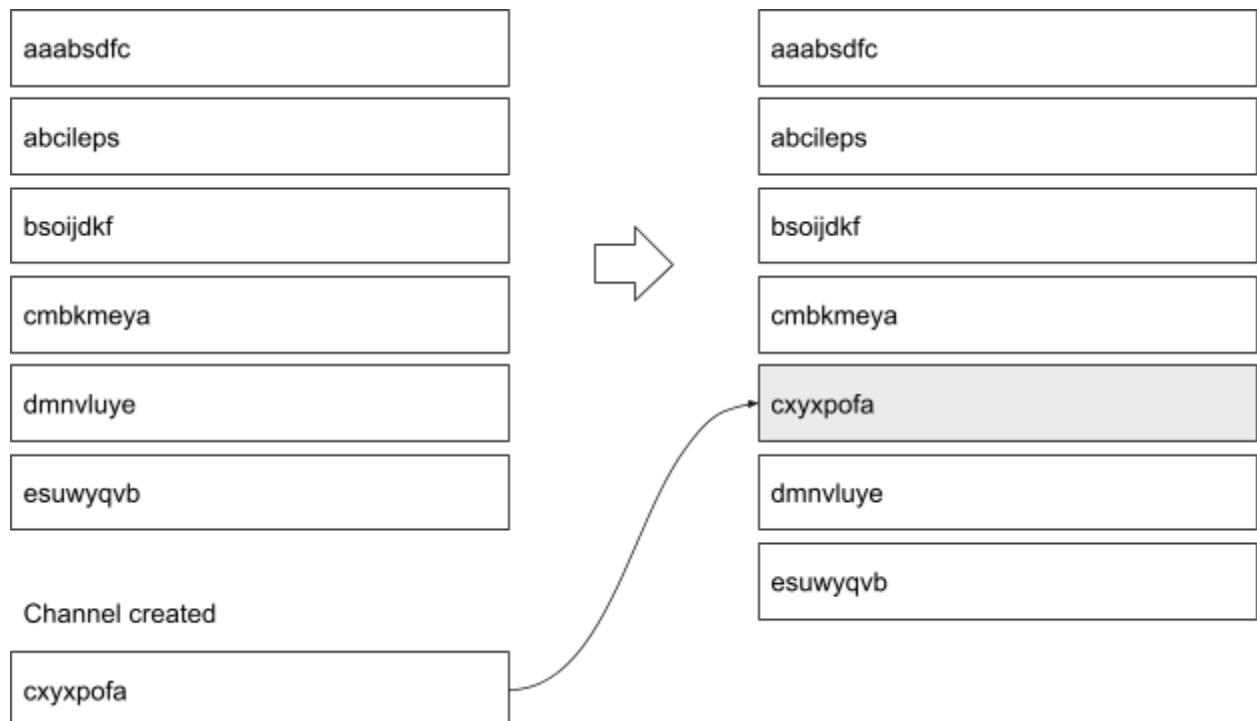


Figure 2. Created channel should be located

Step 3

Limit the height of the list view to be less than the height of 10 items for pagination. Then fetch the channels through the SDK. Replace the dummy data with the fetched channel names. The list view should load next channels when the scroll hits the bottom.

Step 4

Add a click event to the list item. When an item is clicked, update the channel name to the newly generated string of 8-English-letters. Locate the item to the right place after the update. See Figure 3.

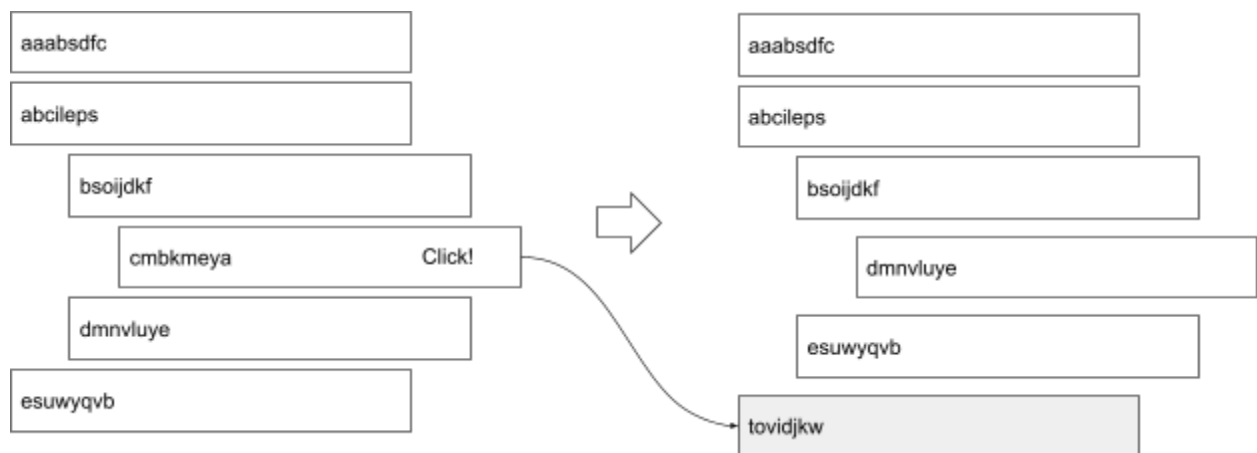


Figure 3. Updated channel should be located